CDF Data Handling System

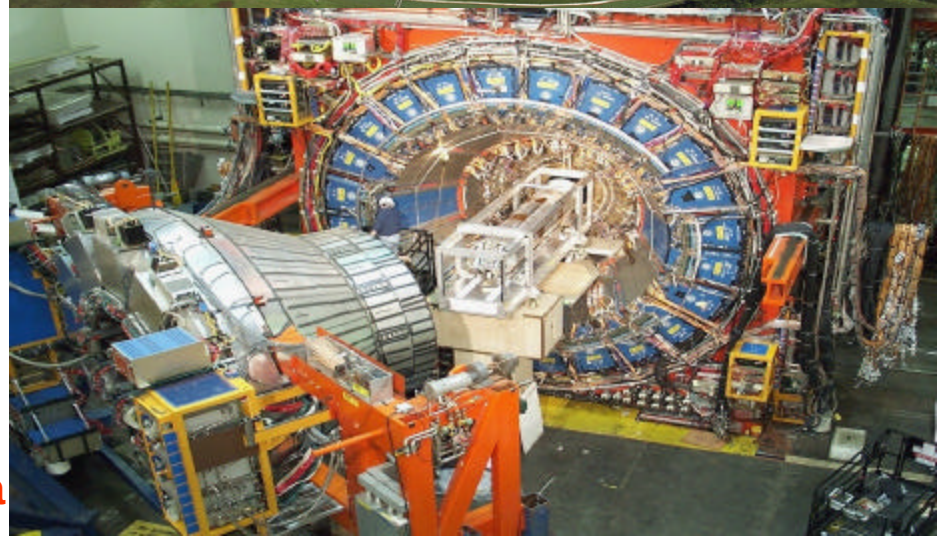
Dmitry Litvintsev

Fermilab CD/CDF for the CDF DH group

- Introduction
- System Components
- System Evolution
- Conclusion

CDF Experiment

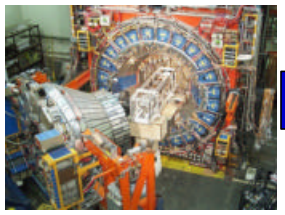
- CDF experiment analyzes data produced in proton-antiproton collisions at Tevatron Collider at Fermilab at c.m.s energy of about 2TeV
- CDF Detector has been upgraded for Run II (March 2001-) for 10x increase in luminosity and 10% increase in energy
- Expected integrated luminosity: 2fb^{-1} (Run IIa), 15fb^{-1} (Run IIb)
- About 600 physicists from 55 institutions from 11 counteis participate in CDF Run II Collaboration



Physics at CDF:

- **Search for Higgs**
- **Precision EWK physics**
- **Top quark properties**
- **QCD at large Q^2**
- **Heavy Flavor Physics**
- **Search for non Standard Model phenomena**

CDF Run II Data Flow (till 05/2002)



DAQ

300 Hz

Level 3 Farm

20 MB/s

75 Hz

Consumer Server/Logger b0dau32

Defines ~50 primary Datasets

Writes events into Files

8 data streams based on dataset bits

B0

Dual Ported Disk

A APHYSR,ACOSMR,ATESTR
B …
C …
D …
E …
F …
G …
H …
I IPHYSR *(inclusive stream)*

CXFS

FCC

20 MB/s

12 TB SCSI RAID arrays

Tape Robot (tape archive) 1PB

FCC Data Logger fcdfsgi1

Organizes: Files into Filesets

Writes: Filesets to Tape

13 MB/s

20 MB/s

GigaBit Switch

fileservers /home /spool

fcdflnx2

fcdflnx3

IO

Production Farm PCs

Concatenates files

Reconstructs Events

Output: ~50 primary Datasets

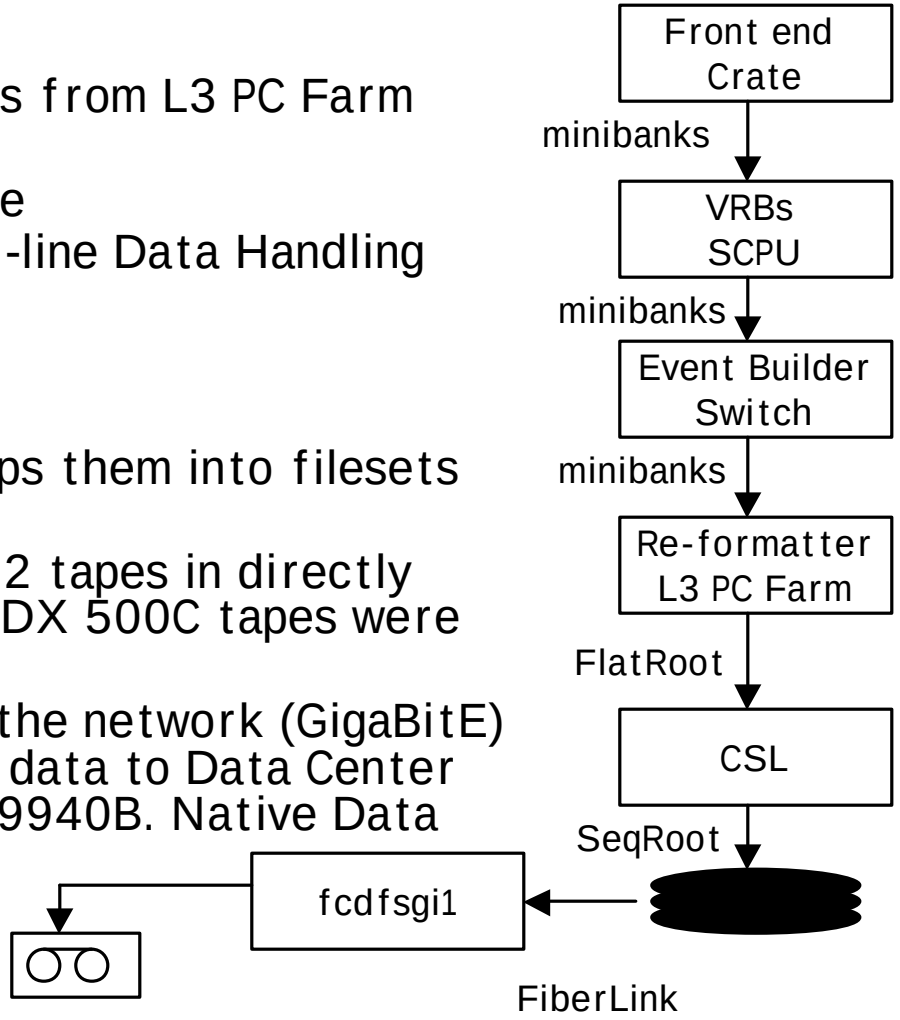
Data Flow (cont)

On-line:

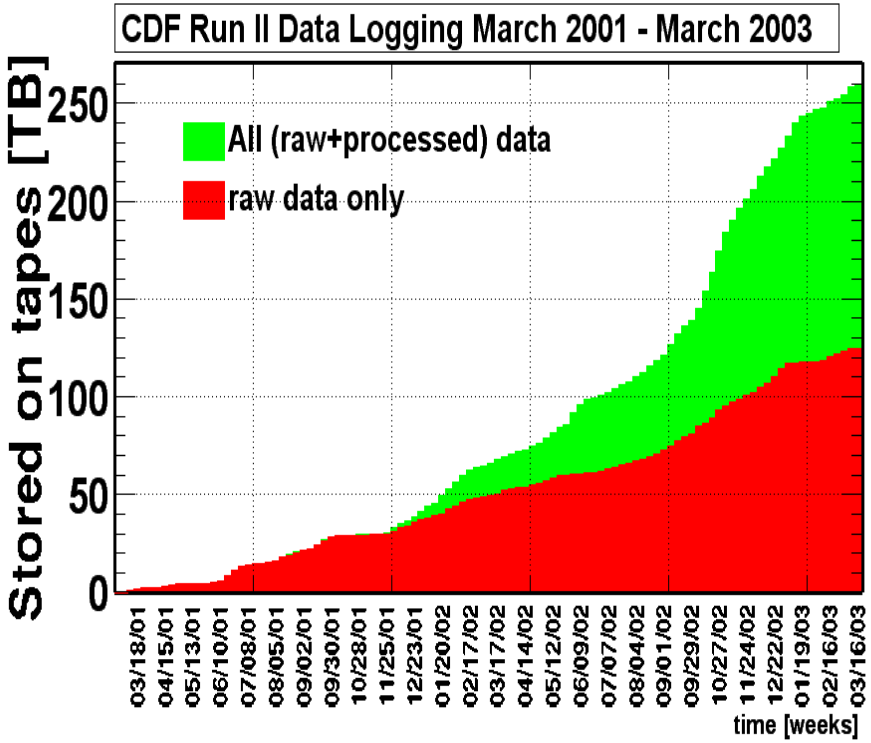
- Consumer Server Logger receives events from L3 PC Farm
- Writes events into ~1GB ANSI files
- Pushes files over FiberLink to FCC where
- From there on files are part of CDF off-line Data Handling System

Off-line:

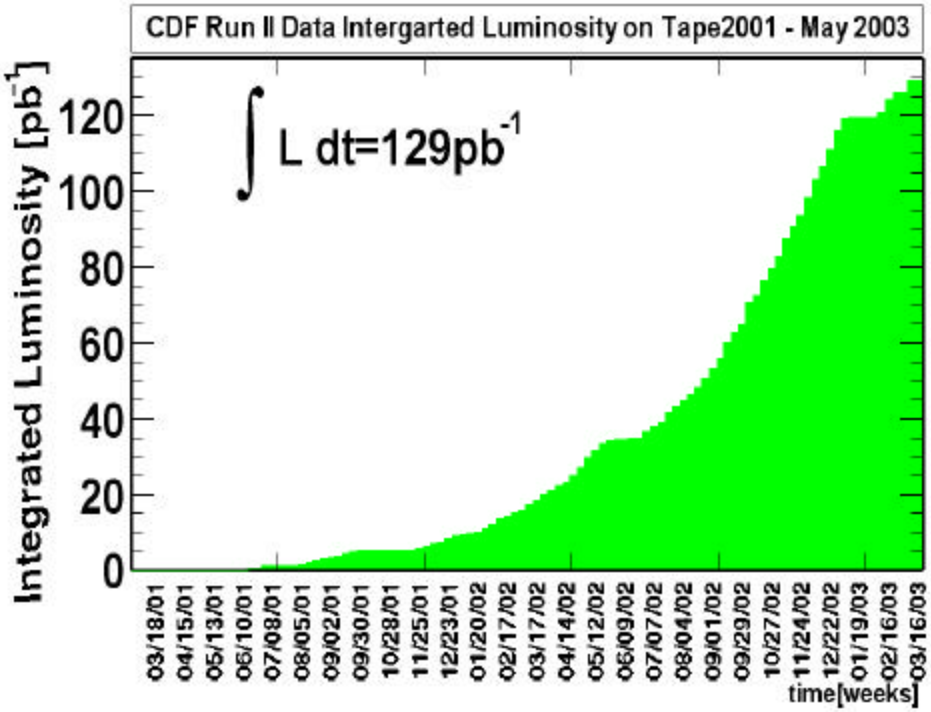
- FilesetTape daemon receives files, groups them into filesets and:
  - Till 05/2002 wrote filesets to AIT-2 tapes in directly attached SCSI tape drives SONY SDX 500C tapes were managed in ADIC AML/2 robot
  - since 05/2002 writes filesets over the network (GigaBitE) into Enstore MSSM. Enstore writes data to Data Center quality drives, STK-T9940A/STK-T9940B. Native Data cartridges. STK-Silo tape library



Data Volumes at CDF



some older produced data was deleted to free up tapes

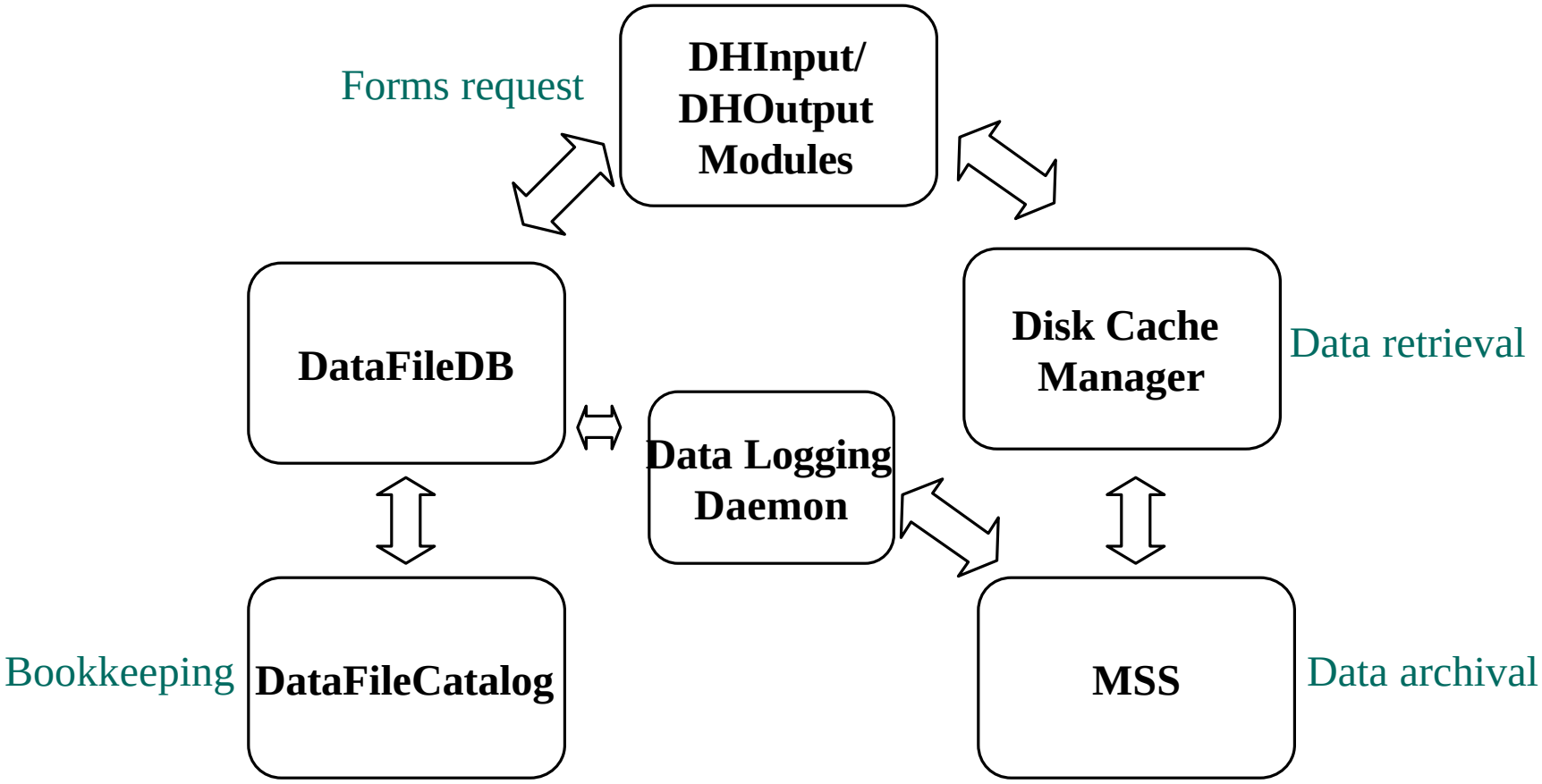


We have collected 6% of expected Run IIa integrated luminosity. And already 120 TB of raw data. This would translate into 2PB of raw data only

DH Overview

•DH System:

Organize, access and archive the data



DH Components/Data Access

- DHInput/DHOutput – high level user interface to DH-system
- DataFileCatalog – Oracle relational DB that stores meta-data information about CDF datasets
- DataFileDB - C++ API that allows to manipulate that data from algorithm code
- Disk Cache Manager in front of HSM:
  - Disk Cache Manager, a.k.a. Disk Inventory Manager – cache layer in front of MSS functioning @ fcdfsgi2 (128 300Mhz MIPS SMP Origin 2000 SGI):
- LSF batch system
- Enstore – generic interface to MSS
- MSS – Robotic tape library – Dual STK Silo Powderhorn 9310 with about 2.2PB capacity data center quaility STK drives – T9940A (being replaced by T9940B)

As more powerful, commodity CPU based, computing facilities – CAF (prototype CAF1 and large scale CAF2 ~600 CPUs) appear users move their analyses there. The data access then is provided by variety of means:

- Via dcap or dccp directly from dCACHE – new Disk Cache Manager featuring network mounted disk read/write pools serving as cache in front of Enstore system (See Rob Kennedy's talk)
- Via rootd from fcdfsgi2 or fileservers running rootd
- Via NFS mounted disks from 'static' fileservers
- Via SAM – an alternative DH system originally developed by D0 collaboration and being implemented by CDF

AC++ DH Modules

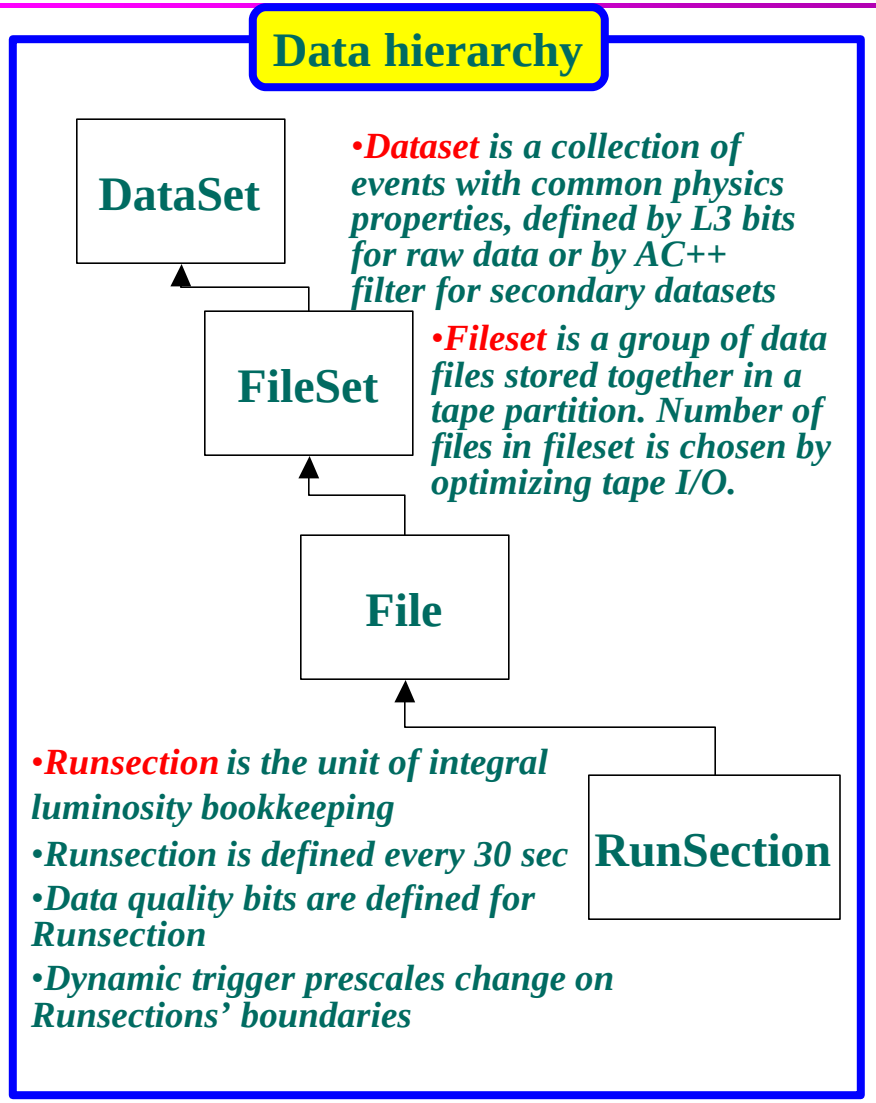
- Jointly with BaBar experiment at SLAC CDF has developed OO analysis framework, an AC++ (as the next generation of AC or Analysis Control of Run I)
- Framework provides hooks to plug in modules that may perform specific tasks. Modules are independent
- DHInput/DHOutput modules provide user friendly interface to DataFileCatalog and and cache manager allowing for seamless access to persistent data. Eg:

```
AppUserBuild::AppUserBuild(AppFramework* frame)
      : AppBuild(frame) {
      frame->add(new DHInputModule());
      frame->add(new DHOutputModule());
      . . .
      frame->add(new MySusySearchModule());
}
```

- DHInput communicates with DataFileCatalog by via DataFileDB API layer and translates user request into list of files to be retrieved from HSSM
- DHInput provides fast navigation through input data made possible by direct assess ROOT format of CDF data
- DHOutput module writes out ANSI files furnished with necessary BOR, Empty Runsections records and makes entries in DataFileCatalog

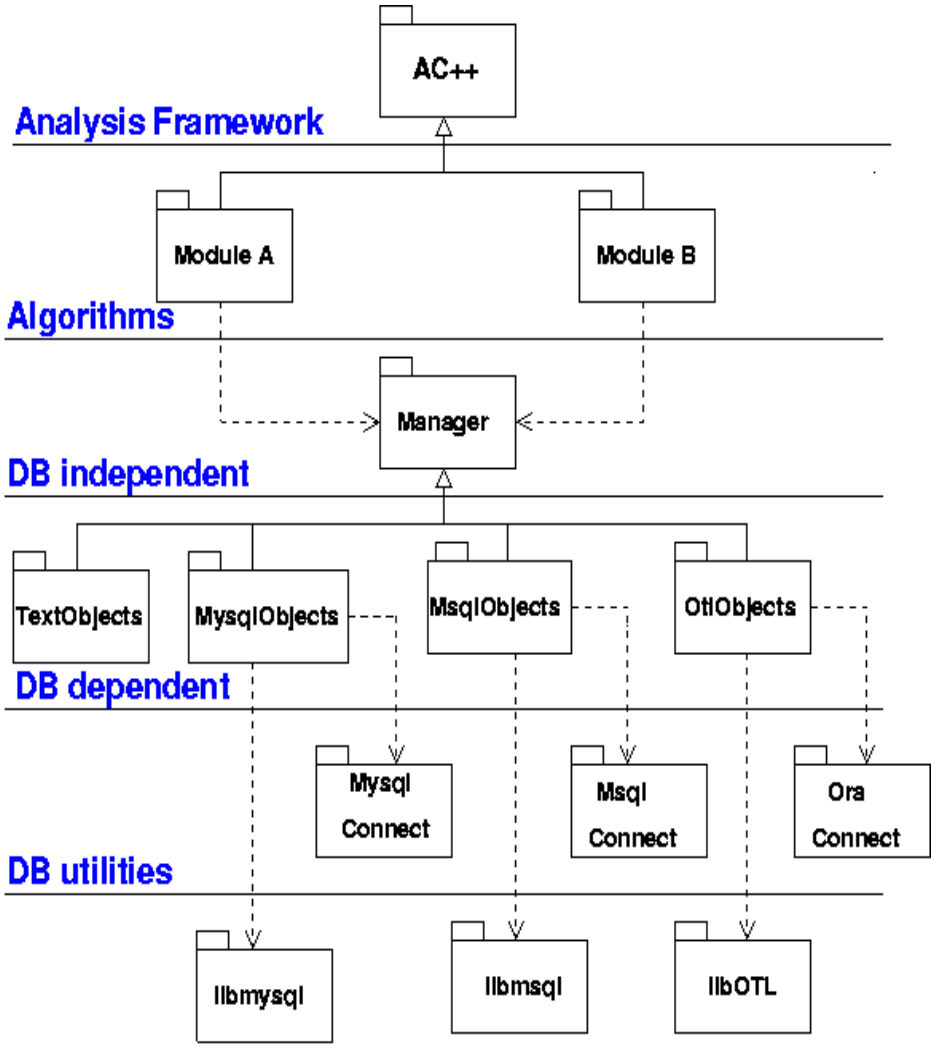
Data File Catalog

- Data File Catalog is a relational DB that contains information about CDF datasets.
- Table structure of DFC reflects hierarchical data organization with RunSections table at the bottom and DataSets table at the top
- DFC allows to store all available CDF meta-data;
  - Raw and centrally produced DST
  - User DST, PAD or Ntuple data
- Central DFC is Oracle, MySQL and mSQL implementations are also supported (used by remote institutions)



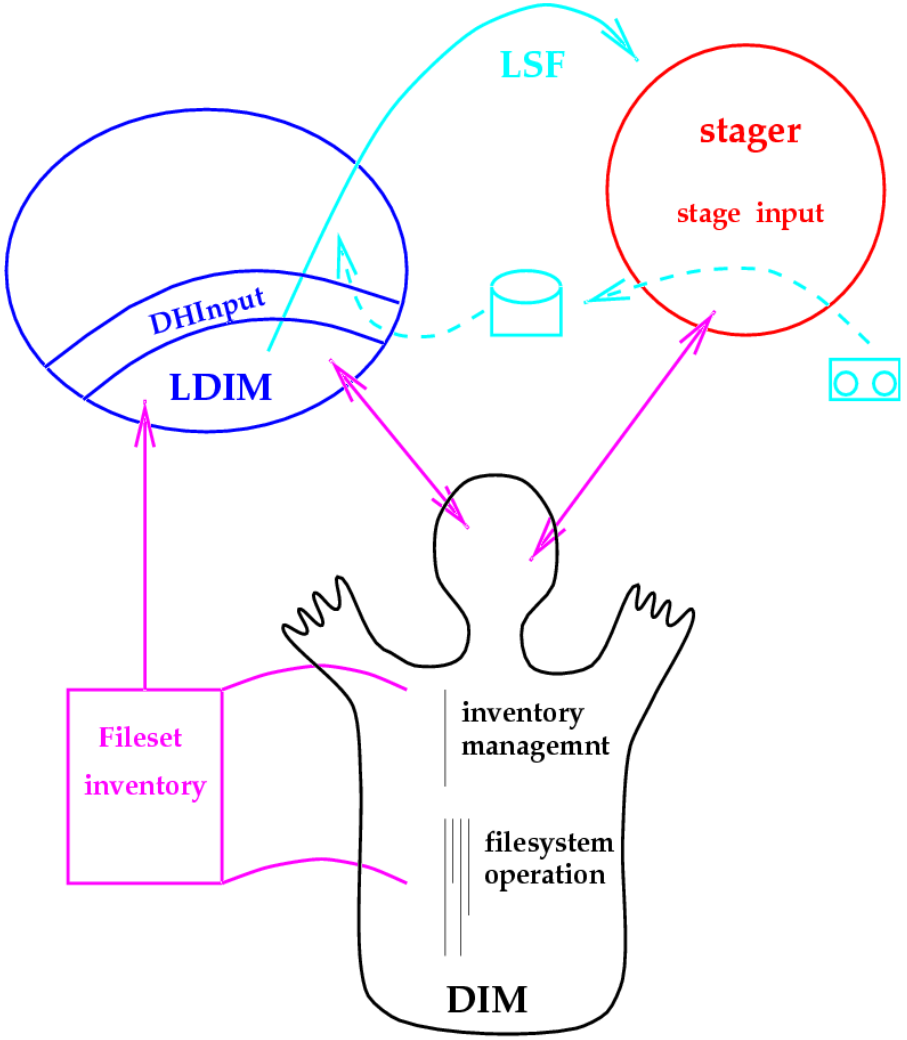
DB Access API

- Data stored in DFC are available to algorithm code via DB independent database management layer
- **DBManager** provides two APIs:
  - Back-end transient to persistent mapping API – ***IoPackage***
  - Template based front-end **Manager<OBJ,KEY>** that provides common **put/get/update/delete** methods on transient objects
- Transient object instances can be cached by key value to configurable depth
- Transient classes definitions, Mapper, Manager<> and Handle<> could be auto generated. Not the case for DFC though
- Problem of keeping Mapper classes in sync for all DB implementations



Resource Manager

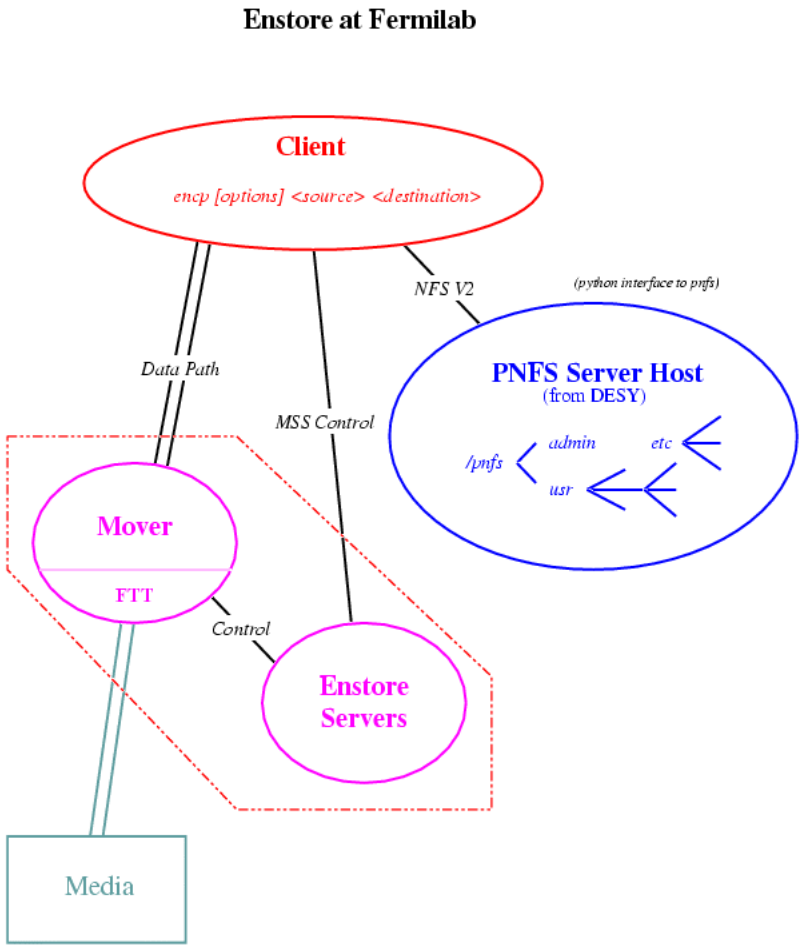
- Disk Inventory Manager acts as cache layer in front of Mass storage system
- User specifies dataset or other selection criteria and DH system acts in concert to deliver the data in location independent manner
- Design choices
  - Client-server architecture
  - System is written in C, to POSIX 10031.c-96 API for portability
  - Communication between client and server are over TCP/IP sockets
  - Decoupled from Data File Catalog
  - Server is multithreaded to provide scalability and prompt responses
  - Server and Client share one filesystem namespace for data directories



CDF Run II MSS

- CDF had started Run IIa experiment with MSS based on cheap commodity AIT-2 tape drives (SONY-CDX500C) directly attached to main CDF data logger (fcdfsgi1) and central analysis SMP (fcdfsgi2). Interface to MSS was based on CDF unique packages.
- This system turned out to be difficult to run smoothly
- As a viable substitute for existing tape system CDF adopted Enstore – generic interface to MSS jointly developed and supported by DESY and Fermilab CD that allows seamless data storage over the network. Main feature of Enstore – network access to the data in the Robot System features:
  - PNFS filesystem – with files being meta data units
  - Request optimization layer
- Data Center quality tape drives STK T9940A/T9940B
- Dual STK-Silo tape robot library

Enstore schematics



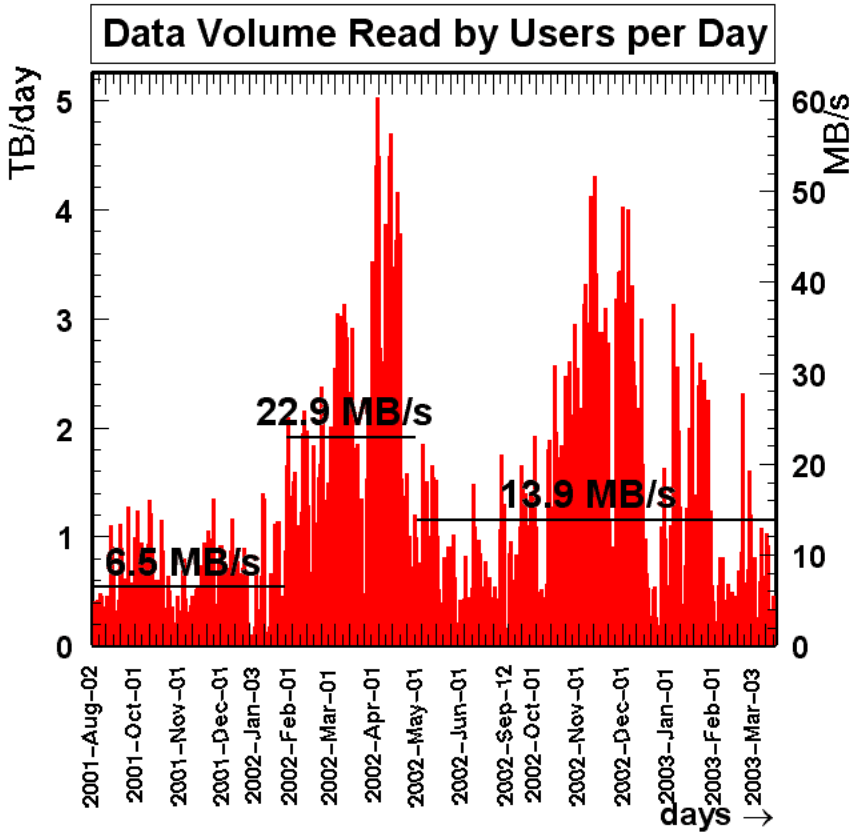
See Don Petravick's talk on FNAL Data Storage Infrastructure

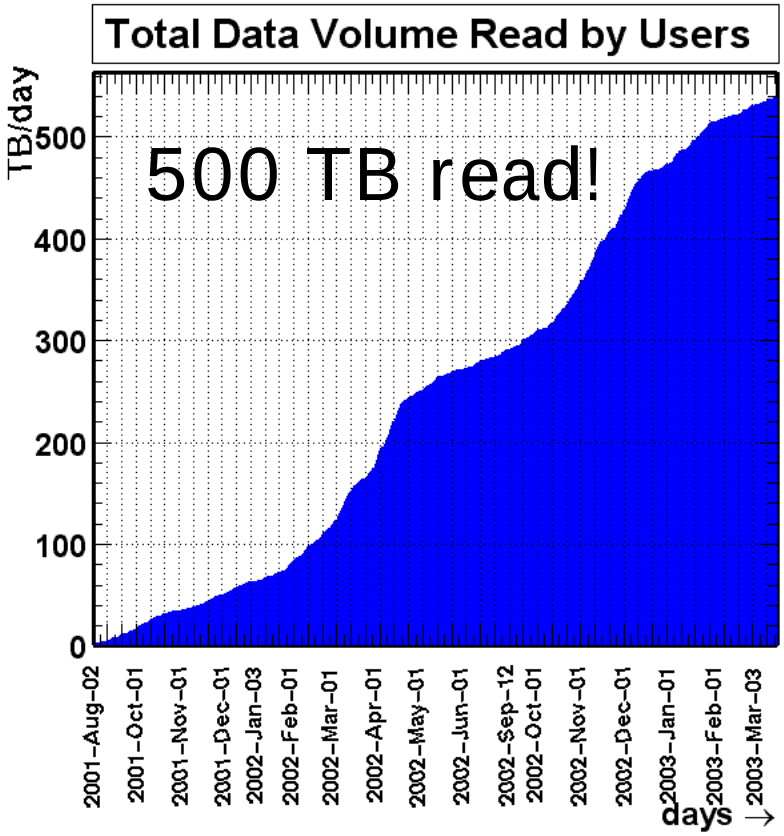
Transition to Enstore

- 100 TB transferred in 3 months using DH system available at fcdfsgi2
- CDF Enstore system is called CDFEN
- no write failures on CDFEN side
- Experience with new system, many issues addressed - 4 versions of 'encp' product in 3 months

Benefits of using Enstore

- Data delivery became stable DH system became robust

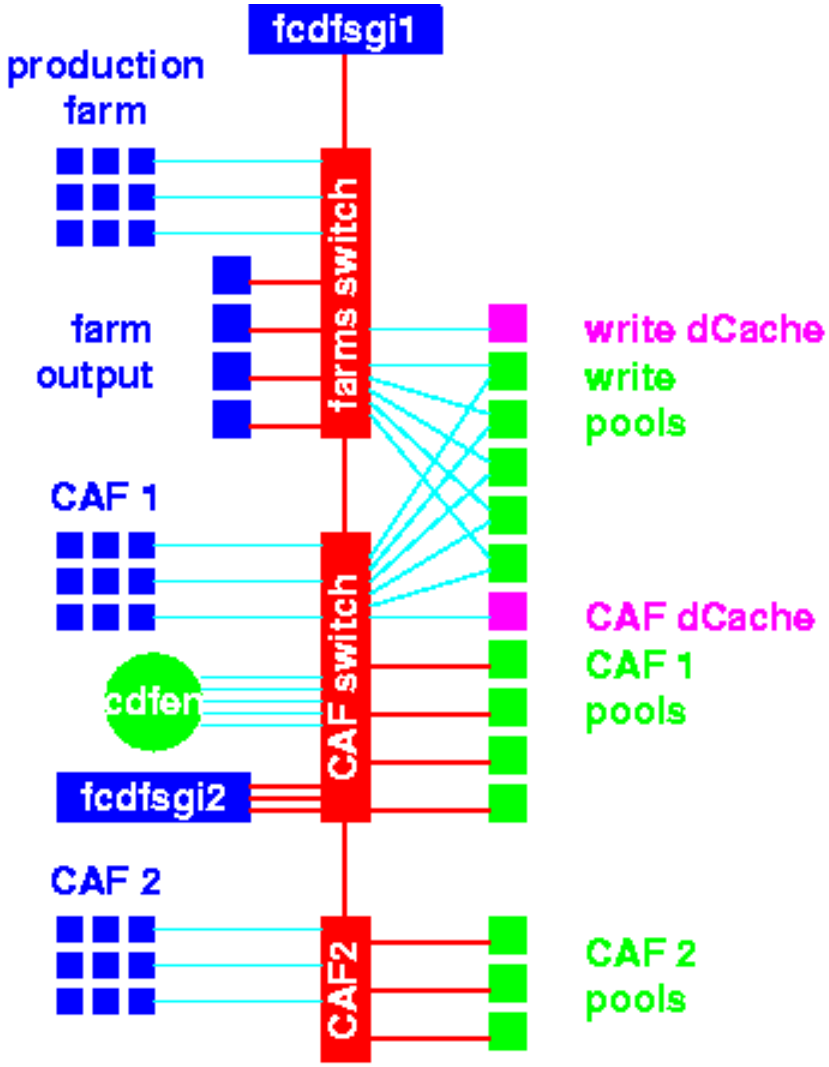




- CDF Production Farm read/write directly from/to Enstore decoupling raw data storage from data delivery to Farms

Current CDF DH Layout

- CDF is adopting the model of distributed computing system. Task of DH is to deliver data to user analysis job running on this system.
- CDF re-evaluated DIM s/w and made a decision to adopt different cache management product dCache which better suited to accommodate distributed systems
  - network attached disk cache for Enstore – read/write disk pools
  - Details are in Rob Kennedy's talk
- About to be declared in production



Conclusion

- Since the start of Run II CDF DH system provided raw data logging and data delivery to user analysis and Central Reconstruction Facilities
- CDF has abandoned use of directly attached commodity tape drives and adopted network mounted data center quality tape drives managed by Enstore system
- CDF has adopted dCache as cache management layer for Enstore. CDF plans to write data directly to dCache making Reconstruction and Analysis tapeless operation!
- Following the changes in CDF computing model CDF DH evolves to provide data to globally distributed computing facilities. CDF is making first steps towards adopting dCACHE based SAM as its main DH system